Social Factors for Data Sparsity Problem of Trust Models in MANETs

M. Shabut, Antesar; Dahal, Keshav

Published in:
Computing, Networking and Communications (ICNC), 2017 International Conference on

DOI:
10.1109/ICCNC.2017.7876247

Published: 13/03/2017

Document Version
Peer reviewed version

Social Factors for Data Sparsity Problem of Trust Models in MANETs

Antesar M. Shabut
Anglia Ruskin University
Chelmsford, UK
antesar.shabut@anglia.ac.uk

Keshav Dahal
University of the West of Scotland
Paisley, UK.
Keshav.Dahal@uws.ac.uk

***Abstract*— The use of recommendation in trust-based models has its advantages in enhancing the correctness and quality of the rating provided by mobile and autonomous nodes in MANETs. However, building a trust model with a recommender system in dynamic and distributed networks is a challenging problem due to the risk of dishonest recommendations. Dealing with dishonest recommendations can result in the additional problem of data sparsity, which is related to the availability of information in the early rounds of the network time or when nodes are inactive in providing recommendations. This paper investigates the problems of data sparsity and cold start of recommender systems in existing trust models. It proposes a recommender system with clustering technique to dynamically seek similar recommendations based on a certain timeframe. Similarity between different nodes is evaluated based on important attributes includes use of interactions, compatibility of information and closeness between the mobile nodes. The recommender system is empirically tested and empirical analysis demonstrates robustness in alleviating the problems of data sparsity and cold start of recommender systems in a dynamic MANET environment.**

Keywords—trust; trust mangement; recommender system; data sparsity; cold start; mobile ad hoc network; MANET

I. INTRODUCTION

Mobile ad hoc network (MANET) is a collection of wireless mobile nodes that are capable of communicating with each other in the absence of a fixed network infrastructure or centralized administration [1]. MANETs are practically emerging as a provider of a flexible method to establish communications in situations where geographical constraints demand a totally distributed system as in hurricane and earthquake disasters, or for exchanging critical information on the battlefield through networking [2]. However, MANET's characteristics, including frequent changes in network topology due to mobility or discontinuous operation of nodes, open wireless medium, and constrained capability, make it vulnerable to security issues in situations where a friendly and cooperative environment is not assumed [3].

Because of the usefulness of the MANET system and its dynamic characteristics, provision of the desired security level for such a system is critical. Researchers have recognized the significance of borrowing trust management concepts from the social network analysis (SNA) field to improve the performance of the network protocols [4]. This move towards social methods in securing MANETs facilitates identification of trust attributes of nodes such as level of cooperation, honesty and pattern of behaviour, to establish and manage trust relationships between nodes in a distributed manner. Trust management technique is one of the approved mechanisms to improve security in MANETs [5]. However, involving the use of recommendation in modelling trust by adopting a recommender system can offer an attractive security mechanism to monitor node behaviour, mitigate attacks, and filter out dishonest nodes.

The goal of recommender systems in MANETs is to provide ratings to every node that might be of interest for other nodes. In particular, recommender systems based on filtering algorithms rely on the opinions expressed by the other nodes. They try to automatically find recommending nodes that are similar to the evaluated nodes and utilize recommendations by these similar nodes. For the MANET, as it is formed by a number of nodes which may be relatively large, recommendations of other nodes are much more sparse than those of general society [6], [7]. This problem can be critical in the establishment of sufficient trust relationships among nodes in MANET. A cold start problem when the recommender systems fail to find sufficient similar neighbors in sparse data is considered as the most challenging problem for recommender systems. Therefore, trust as a social relationship is emerging as an effective method for secure and effective neighbor selection [8], [9]. Although trust has been explored by many researchers in the past as a successful solution for assisting recommender systems [10], issues such as data sparsity and dishonest recommendation problems have not been explored adequately.

This paper investigates the problems of data sparsity and cold start problems associated with recommender systems in existing trust models. We propose a recommender system with a filtering algorithm to filter out dishonest recommendations while mitigating the problem of data sparsity. It utilizes a clustering technique to dynamically seek similar recommendations based on a certain timeframe. Similarity between different nodes is evaluated based on important attributes includes use of interactions, compatibility of information and closeness between the mobile nodes. Such attributes are used to alleviate the data sparsity and cold start problems. The remaining parts of this paper are organized as follows. Section II gives an overview of related work. Section III outlines the proposed model. Section IV presents the simulation results. The conclusions of the paper is provided in Section V.

II. Related Work

Recently, different trust and reputation models have been proposed for routing security purpose in MANETs using direct rating or recommender systems to evaluate trustworthiness of nodes in the network. Trust and reputation models have been widely used by several literatures to enhance the data sparsity and cold start problems of recommender systems [8], [10]–[13]. Since a node can hold a direct opinion on a small number of nodes, predicting the trustworthiness of other unknown nodes can be possible by using different trust metrics and reputation techniques. Propagation of trust relationships over the considered environment is achieved by publishing direct trust information frequently or by request when needed and making it available in the network, and this is critical in solving the data sparsity and cold start problems. A trust metric, which is considered as an algorithm able to propagate trust relationships over the trust network, to filter and aggregate these, and to estimate a dynamic weight to avoid data sparsity problems, has been recently used by different proposed recommendation based trust models [8].

Two studies [11], [12] incorporate the concept of trust into recommender systems to synthesize recommendations based on trust opinion derived from trustworthy entities. For distributed systems such as MANETs, the dependence on generic collaborative filtering algorithms is not effective in mitigating the recommender systems challenges of dishonest recommenders, data sparsity and cold start problems. These algorithms require an effective approach to allow some filtering techniques and formation of neighborhood relationships.

In an attempt to provide high-quality recommendations in the presence of sparse data, one study [13] proposes an algorithm for alleviating sparsity of data by utilising trust inferences. Trust inferences can be defined as the transitive characteristic of trust that represents the associations between participants in the context of an underlying social community. They are considered by the authors as valuable sources of additional information which may help alleviating the sparsity and cold start problems. Their experimental results indicate that the utilisation of trust inferences can significantly improve the quality performance of the classic recommender systems.

Similarly, authors in [8] propose a recommender system technique that depends on the use of trust to alleviate the data sparsity problem which is called as Trust-aware Recommender System. The proposed algorithm is designed to search for trustable users by exploiting trust propagation over the network and not only to search for similar users as most recommender systems do. They evaluate their algorithm by comparing it with different algorithms, including traditional recommender systems that utilize only similarity, recommender systems that utilize only trust information, and recommender systems that combine both trust and similarity. Their empirical results indicate that trust is very effective in alleviating recommender systems' weaknesses related to data sparsity and cold start problems.

Other authors [10] utilize trust to assist their proposed recommender system by extending the number of relationships between participants (i.e. the neighboring base), in which participants can cooperate with more neighbors than in traditional recommender systems and consequently gain access to more recommendations. They use similarity relationships to discover more participants not necessarily directly linked with each other by the assistance of friends who consider those participants as trustworthy. The friends-of-friends concept to discover participants from new users who may hold valuable experience that can be delivered as recommendations about some entities which are of interest to other entities has been used to solve the problem of cold start with sparse recommendation. Their experiment also shows better result in alleviating the cold start problem than a system which makes no use of trust relationships.

In all the above literature, trust is used to alleviate only the problem of the data sparsity of the recommender systems by considering the trust relationships between entities, propagation, aggregation, and transitive properties of trust. However, the use of multidimensional properties of trust to alleviate both data sparsity and dishonest recommendations as weaknesses of the recommender is not considered.

III. The Proposed Model

This paper makes use of a Bayesian statistical approach similar to that used in [14] for computing trust values based on the assumption that they follow a beta probability distribution. Beta distribution is estimated by using two parameters (α, β). These can be calculated by accumulating observations of positive interactions α (forwarding packets) and negative interactions β (dropping packets). This function is defined by gamma function as in Eq. (1),

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, p^{\alpha-1}\,(1-p)^{\beta-1} \qquad (1)$$

where $0 \leq p \leq 1, \alpha, \beta > 0$ with a condition that $p \neq 0\ if\ \alpha < 1\ and\ p \neq 1$ if $\beta < 1$. Gamma function is defined by the integral: $\Gamma(x) = \int_0^\infty e^{-t}\, t^{x-1}\, dt$.

The proposed model is based on a trust monitoring technique, in which nodes monitor each other's behaviors in the performed interactions to develop a direct opinion of each other. A recommender system which is based on a filtering algorithm to filter out the opinions expressed by the other nodes is also proposed. The data sparsity problem is considered, as in MANETs, to be that recommendations of other nodes are far sparser than those of general society. An imputation technique for missing information is used to fill in this information. Propagation of trust relationships over the MANETs environment by publishing direct trust information frequently or by request when needed and making it available in the network is considered to solve data sparsity and cold start problems. The model is consisted of different components which is described below.

A. Trust computation component

Direct Trust T_{ij}^{direct}: In MANETs, direct trust is obtained when two nodes have already initiated a trust relationship and

they can immediately interact with each other (at least for a specific period of time, when they are within the same range, because of nodes' mobility), without requiring a third-party opinion or recommendation. It is calculated as in Eq. (2),

$$T_{ij}^{direct} = \frac{\alpha_{ij}}{\alpha_{ij} + \beta_{ij}} \quad (2)$$

where α_{ij} represents the accumulated positive interactions between node i and j while β_{ij} represents the accumulated negative interactions between node i and j.

Indirect Trust $T_{ij}^{indirect}$: Indirect trust needs to be considered when two nodes have not established a previous trust relationship through exchange of packets or any other form of communication. In such cases, the evaluating node does not have sufficient experience to judge the trustworthiness of the other node being evaluated. Indirect trust is calculated similarly to the way in which direct trust was computed earlier. It is calculated according to Eq. (3),

$$T_{ij}^{indirect} = \sum_{k=1}^{N} \frac{\alpha'_{kj}}{\alpha'_{kj} + \beta'_{kj}} \quad (3)$$

where i is the evaluating node, N is the number of recommendations, j is the evaluated node, while k is the recommending node, α'_{kj} represents the accumulated positive interactions between node k and j and β'_{kj} represents the accumulated negative interactions between node k and j.

The proposed model incorporates a decay factor μ to gradually decrease the influence of past experience over time, prior to aggregation with new trust values. The first situation given is when a node observes an additional new positive or negative interaction between time t_c and t_{c+1} denoted as α^{new} and β^{new}. In this case, the updated α and β should be reduced by the decay factor μ before merging them with the new values. Therefore, at time t_{c+1}, α and β are updated respectively according to the formula in Eq. (4),

$$\alpha = \alpha^{old} * \mu + \alpha^{new} \ , \ \beta = \beta^{old} * \mu + \beta^{new} \quad (4)$$

where $0 \leq \mu \leq 1$. The second situation is when there is no observed new positive and negative interaction between time t_c and t_{c+1}. Then, at time t_{c+1}, α and β are updated respectively as in Eq. (5).

$$\alpha = \alpha^{old} * \mu \ , \ \beta = \beta^{old} * \mu \quad (5)$$

Trust Value T_{ij}: For each node in the network, trust value T_{ij} is calculated by combining direct and indirect trust values with different weights, denoted by w_{direct} and $w_{indirect}$ respectively. Trust value T_{ij} is computed according to Eq. (6),

$$T_{ij} = w_{direct} T_{ij}^{direct} + w_{indirect} T_{ij}^{indirect} \quad (6)$$

where $0 \leq w_{direct} \leq 1, 0 \leq w_{indirect} \leq 1,$ and $w_{direct} + w_{indirect} = 1$.

B. Recommender System Component

The recommender system component in the proposed model requests and gathers recommendations for a node from a list of recommending nodes. It helps in detecting and eliminating false recommendations in sparse data. The recommender system requests and gathers a recommendation list for an evaluating node i about node j from a list of recommending nodes $\{k_1, k_2, k_3, \dots, k_N\}$ between time t_i and $t_{i+1}$and sends it to the cluster system to run the filtering algorithm. After filtering, it receives the trustworthy clusters as a list of honest recommendations denoted as $\{k_1^{Tr}, k_2^{Tr}, k_3^{Tr}, \dots, k_N^{Tr}\}$. The final task is to send the trustworthy cluster $C^{Trustworthy}$ to the requesting node. Algorithm 1 illustrates the recommendation manager algorithm.

Algorithm 5-1: Recommendation Manager Algorithm

1. ***For*** *each recommendation request* ***Do***
2. ***Send*** *request to neighbors*
3. ***Collect*** *received recommendation*
4. ***Construct*** $L = \{k_1, k_2, k_3, \dots, k_N\}$
5. ***Send*** *L to the cluster manager for processing*
6. ***Receive*** *trustworthy cluster* $C^{Trustworthy} = \{k_1^{Tr}, k_2^{Tr}, k_3^{Tr}, \dots, k_N^{Tr}\}$
7. ***Send*** $C^{Trustworthy}$ *to the requested node*
8. ***End*** *For*

C. Cluster System Component

The proposed trust model uses a clustering technique in order to maximize consistency in receiving recommendations. Recommendations from a misbehaving node can have a range of multiple different ratings for the evaluated node, besides, by increasing the network rounds, the ratings may become too sparse. Therefore, finding similar ratings is challenging for traditional recommender systems. The dynamic clustering of recommendations over a period of time proposed in this paper can filter out deviating ratings from the list of recommendations and alleviate the data sparsity and cold start problem. The clustering system is based on four attributes namely, *time frame*, *confidence*, *deviation*, and *closeness centrality* which described in the following subsections.

Time frame: The proposed filtering algorithm takes into consideration the dynamic characteristics of MANETs which change over time. The algorithm divides the time frame into three important periods: past, current time and future. The model uses the past period, which represents experiences collected and accumulated by nodes using previous interactions. The present period represents the time when nodes use the model to predict ratings. Meanwhile, the future period represents the time for predicting the future ratings using the received recommendations. As the model uses past experiences to predict the future ratings of nodes, it is important to divide the past as well into distant past, which represents old information, and the recent past, which represents the recent ratings, using a dynamic time period according to the number of recommendations. The model uses Eq. (4) and (5) to forget about the distant past.

Confidence Value V_{ij}^{conf}: Basically, confidence value is utilized in the proposed trust model to solve the problem of short-term and long-term observations. That is, nodes may have the same level of trust with different numbers of observations. For example, nodes in the network can have nearly the same level of trust although they may have different levels of observations. Consequently, this can lead to wrong estimation in the recommender system. The nodes with higher confidence value (those having sufficient interactions with the evaluated node) are desirable because the higher number of interactions will offer rich information which would help in selecting better recommending nodes. Therefore, the confidence value can be used to solve the problem of data sparsity and enhance the recommender system ability to predict the correct ratings and reduce errors. The confidence value is computed as in Eq. (7).

$$V_{ij}^{conf} = 1 - \sqrt{12}\sigma_{kj}$$
$$V_{ij}^{conf} = 1 - \sqrt{\frac{12\,\alpha_{kj}\beta_{kj}}{\left(\alpha_{kj} + \beta_{kj}\right)^2\left(\alpha_{kj} + \beta_{kj} + 1\right)}} \quad (7)$$

Deviation Value V_{ij}^{dev}: Deviation value represents to what extent the received recommendation is compatible with the personal experience of evaluating node. This value has been used by the means of the deviation test in [16] to ensure the unity of view with the receiving node. Each node compares received recommendation with its own first-hand information and accepts only those not deviating too much from self-observations. To ensure correctness of this value, deviation test is only applied if both nodes have similar level of confidence. Assume that there are three nodes (i, j and k), and node i attempts to calculate the trust value of its neighbor node j using recommendation provided by node k, deviation value is calculated as follows:

$$V_{ij}^{dev} = \left|T_{ij}^{d} - T_{kj}^{r}\right| \leq d^{dev} \quad (8)$$

where T_{ij}^{d} is the direct trust value of i about j, and T_{kj}^{r} is the received trust value of k about j, d^{dev} is the deviation threshold.

Closeness Centrality Value V_{ij}^{close}: Closeness centrality measures the distance between the evaluated node and the recommending node in terms of physical distance, number of hops, or delays. In the proposed model, closeness centrality is a measure of the distance between the evaluating node and the recommending node. The use of closeness centrality enhances the filtering algorithm as close friends may have more interactions in the time of the friendship. Consequently, the use of closeness as a similarity attribute can help alleviate the data sparsity problem as rich information would be available. It is calculated by Eq. (9).

$$V_{ij}^{close} = \sqrt{(x_i^{loc} - x_k^{loc})^2 + (y_i^{loc} - y_k^{loc})^2} \leq d^{dis} \quad (9)$$

where (x_i^{loc}, y_i^{loc}), (x_k^{loc}, y_k^{loc}) are the positions of node i and node k at time t and d^{dis} is a predefined distance threshold between node i and node k which should be less than the transmission range.

The filtering algorithm uses the clustering process with the four factors explained above to merge ratings with the closest similarity. Afterwards, it selects the trustworthy clusters if all of the recommending nodes in a specified cluster satisfy predefined rules. The next step is to apply majority rule to select the cluster with the largest number of members. In the final step, trustworthy clusters are returned to the recommender system and to the evaluating node to predict the trustworthiness of the evaluated node

IV. Simulation and Results

NS2 simulator [17] is utilized to test the validity of the proposed filtering algorithm to alleviate the problem of data sparsity and cold start using simulation. For details of the attack model used in this experiment and the honesty of the generated recommendations by the proposed algorithm, we refer the reader to our previous paper in [15]. Table I shows the network configuration.

TABLE I. NETWORK CONFIGURATION TABLE

Parameter	Value	Parameter	Value
Nodes	50	Packet size	512 B
Area	700 m X 700 m	Application	CBR
Source-destination pairs	15	Transmitting capacity	2 Kbps
Radio Range	250 m	Simulation time	500 s
Movement	Random waypoint model	Trust threshold	0.4
Speed	10 m/s	Fading timer μ	10 s
Routing Protocol	DSR	d^{dev}	0.5
MAC	802.11	d^{dis}	200 m

Fig. 1 demonstrates the percentage of nodes with recommendation for node 15 based on time of simulation. The x-axis in Fig. 1 shows the simulation time, while y-axis displays the percentage of nodes with information about node 15. A comparison has been made between two different recommender systems as follows. First, the traditional recommender systems which use only one recommendation at the time of requesting or publishing recommendations and uses the only metric, namely the deviation threshold, to filter recommendations [16]. Second, the proposed recommender system with the filtering algorithm uses the clustering technique for a certain timeframe based on three properties of trust. It can be seen that with increasing time of simulations, the percentage of nodes having recommendations about node 15 increases for both the traditional recommender systems and for the proposed recommender with the filtering algorithm, but also that the percentage of nodes with information about node 15 in the proposed recommender system with the filtering algorithm is always greater than for the traditional recommender systems: especially in the early rounds of the network when information is very limited due to fewer nodes having interactions with node 15. This figure shows that the filtering algorithm is capable of alleviating the cold start problem in the early rounds of the network when less

information is available and the similarity concept is difficult to apply.

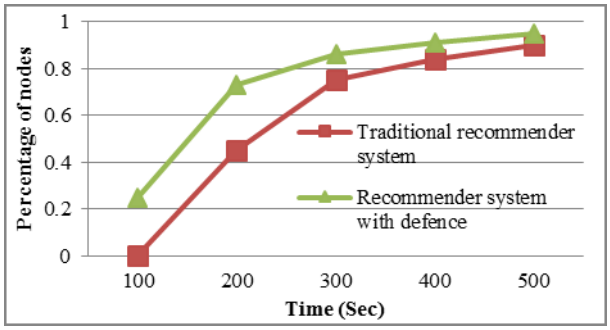

Fig. 1. Percentage of node recommendation of node 15 based on time of simulation

To test the effectiveness of the proposed filtering algorithm further in alleviating the data sparsity problem, another additional metric is defined: Mean Absolute Error (MAE), which representing a measure of the deviation of the predicted recommendations from their actual correct values. The average of all the MAE for every single node in the network is computed, and the result is shown in Fig 2. As in Fig 1, a comparison is conducted between two of the traditional recommender systems and the proposed recommender system with the filtering algorithm. The figure shows that the MAE of the proposed filtering algorithm is always less than the traditional recommender systems for MANETs over the entire time of the simulation. The Mean Absolute Error convergence is reduced to the smallest value, of nearly zero, when the time of simulation reaches 500 seconds, because of the ability of the proposed filtering algorithm to correctly predict ratings, and its reduction in error in the prediction process to the smallest values while the traditional algorithm produces an error of nearly 0.15 in the same time. The results show that the filtering algorithm is capable of alleviating the problem of data sparsity in recommendations when most nodes are active in providing recommendations.

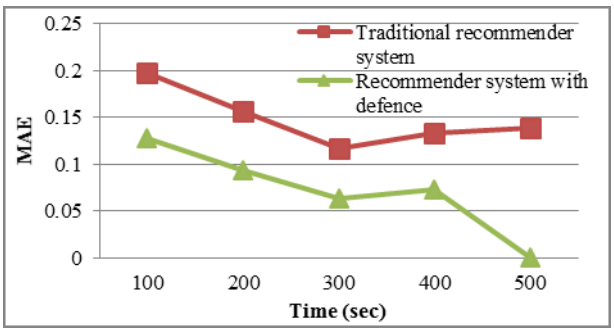

Fig. 2. Mean Absolute Error (MAE) of the predicted recommendation

V. CONCLUSIONS

A recommender system with a filtering algorithm in a trust-based model has been proposed to enhance the correctness and quality of the rating provided by mobile and autonomous nodes in MANETs. Besides this, it alleviates the problems of data sparsity and cold start of recommender systems in existing trust models using trust properties of propagating, inferring, and aggregating trust relationships over a distributed environment such as MANETs. The proposed algorithm utilizes clustering technique to dynamically seek similar recommendations based on multidimensional attributes. The proposed filtering algorithm shows that it is able to alleviate the data sparsity problem as well as filtering dishonest recommendations.

REFERENCES

[1] H. Deng, W. Li, and D. Agrawal, "Routing security in wireless ad hoc networks," IEEE Commun. Mag., 2002.

[2] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," Ad Hoc Networks, vol. 1, no. 1, pp. 13–64, 2003.

[3] H. Hao Yang, H. Haiyun Luo, F. Fan Ye, S. Songwu Lu, and L. Lixia Zhang, "Security in mobile ad hoc networks: challenges and solutions," IEEE Wirel. Commun., vol. 11, no. 1, pp. 38–47, Feb. 2004.

[4] D. Katsaros, N. Dimokas, and L. Tassiulas, "Social network analysis concepts in the design of wireless Ad Hoc network protocols," IEEE Netw., vol. 24, no. 6, pp. 23–29, Nov. 2010.

[5] W. Li, J. Parker, and A. Joshi, "Security Through Collaboration and Trust in MANETs," Mob. Networks Appl., vol. 17, no. 3, pp. 342–352, Jun. 2012.

[6] J. Luo, X. Liu, and M. Fan, "A trust model based on fuzzy recommendation for mobile ad-hoc networks," Comput. Networks, vol. 53, no. 14, pp. 2396–2407, 2009.

[7] K. Ren, T. Li, Z. Wan, F. Bao, R. H. Deng, and K. Kim, "Highly reliable trust establishment scheme in ad hoc networks," Comput. Networks, vol. 45, no. 6, pp. 687–699, 2004.

[8] P. Massa and P. Avesani, "Trust-aware recommender systems," in Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07, 2007, p. 17.

[9] P. Massa and P. Avesani, "Trust Metrics in Recommender Systems," Springer London, 2009, pp. 259–285.

[10] G. Pitsilis and S. Knapskog, "Social Trust as a solution to address sparsity-inherent problems of Recommender systems," arXiv Prepr. arXiv1208.1004, 2012.

[11] M. Kinateder and K. Rothermel, "Architecture and Algorithms for a Distributed Reputation System," Springer Berlin Heidelberg, 2003, pp. 1–16.

[12] M. Montaner, B. López, and J. L. de la Rosa, "Opinion-Based Filtering through Trust," Springer Berlin Heidelberg, 2002, pp. 164–178.

[13] M. Papagelis, D. Plexousakis, and T. Kutsuras, "Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences," Springer Berlin Heidelberg, 2005, pp. 224–239.

[14] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," Decis. Support Syst., vol. 43, no. 2, pp. 618–644, 2007.

[15] A. M. Shabut, K. P. Dahal, S. K. Bista, and I. U. Awan, "Recommendation Based Trust Model with an Effective Defence Scheme for MANETs," IEEE Trans. Mob. Comput., vol. 14, no. 10, pp. 2101–2115, Oct. 2015.

[16] S. Buchegger and J.-Y. Le Boudec, "A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks," P2PEcon 2004, 2004.

[17] T. Issariyakul and E. Hossain, Introduction to network simulator NS2. 2011.